ARM-based Hardware Requirements Specification

Palm OS® 5 PDK/SPK

PalmSource Confidential

CONTRIBUTORS

Written by Dennis Parker

ARM-based Hardware Requirements Specification
Document Number 5048-001
April 12, 2002

PalmSource, Inc.
5470 Great America Pkwy.
Santa Clara, CA 95054
USA
www.palmos.com

Table of Contents

About This Document

ARM-based Hardware Requirements Specification is a guideline which licensees can use to assist in developing products based on Palm OS® 5 running on ARM-based hardware. This document describes the typical hardware components and the characteristics of those components that are required to design an ARM-based product that will function with the Palm OS software.

What this Document Contains

This section provides an overview of the chapters in this document:

- Chapter 1, "Introduction." Answers frequently asked questions about Hardware Requirements Specification (HRS) and provides an overview of a system based on the HRS.
- Chapter 2, "System Details." Describes the characteristics of each of the major system components.
- Appendix A, "Supported Flash Devices." Describes how to write a flash nub so that you can use the 68K Palm Debugger to flash ARM ROMs.
- "ARM HRS Glossary." Describes ARM-related hardware terms.

Related Documentation

The following documentation should be used in conjunction with the ARM-based Hardware Requirements Specification (HRS).

Document	Description
Customizing the Palm OS Platform	This manual provides information on customizing Palm OS using ARM-native code. Discussion includes writing ARM shared libraries, integrating ARM code with 68K applications, and creating OS patches.
DAL Reference	This manual is a companion to the *DAL Customization Guide*. It describes the API routines in the Hardware Abstraction Layer (HAL), the kernel Hardware Abstraction Layer (kHAL), and the Kernel Abstraction Layer (KAL) . These routines serve two purposes. They are either modifed by you to accomodate specific hardware features, or called to accomplish a particular task.
ARM7TDMI documentation, http://www.arm.com	Full range of ARM application notes, user guides, technical specifications, white papers, and glossary.

Additional Resources

- Training

  Palm and its partners host training classes for Palm OS developers. For topics and schedules, check

  http://www.palmos.com/dev/training

- Knowledge Base

  The Knowledge Base is a fast, web-based database of technical information. Search for frequently asked questions (FAQs), sample code, white papers, and the development documentation at

  http://www.palmos.com/dev/support/kb/

1

Introduction

This chapter answers frequently asked questions about the Hardware Requirements Specification (HRS) and provides an overview of a system based on the HRS.

What Problem Does the HRS Solve?

The HRS establishes a baseline, somewhat generic design that is known to work with Palm OS® software. Use this reference design as the starting point when designing devices that vary from the baseline. It restricts the possible problems to a known set–the differences between the reference design and the target design.

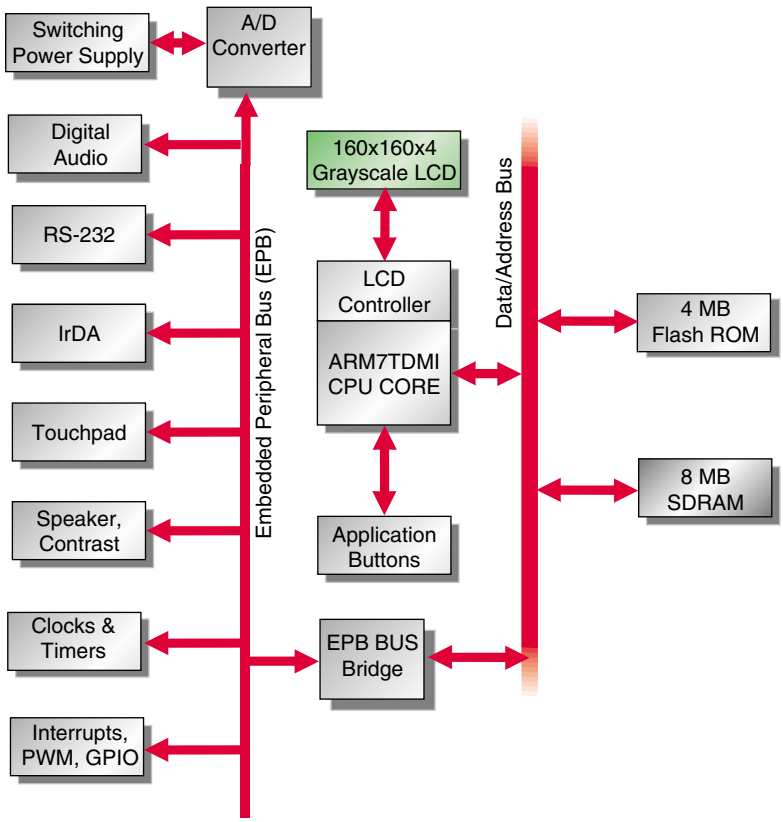
When Do You Use the HRS?

You use the HRS when designing a device based on the Palm OS software to ensure that the device is implemented in a way that will be supported by the operating system.

Requirements Overview

Figure 1.1 illustrates a typical device with a monochrome display based on this Palm OS software release.

Figure 1.1 Typical Device Configuration - Monochrome Display



The device illustrated includes the following hardware components:

- ARM7TDMI microprocessor which includes an LCD controller
- Grayscale LCD (160x160 pixels)
- Flash ROM (4 MB) for storing system software and built-in applications
- SDRAM (8 MB) for storing additional applications and user data
- RS-232 and IrDA input/output interfaces
- Touchpad for pen input
- Speaker and contrast control
- Application buttons to wake up the system and launch applications

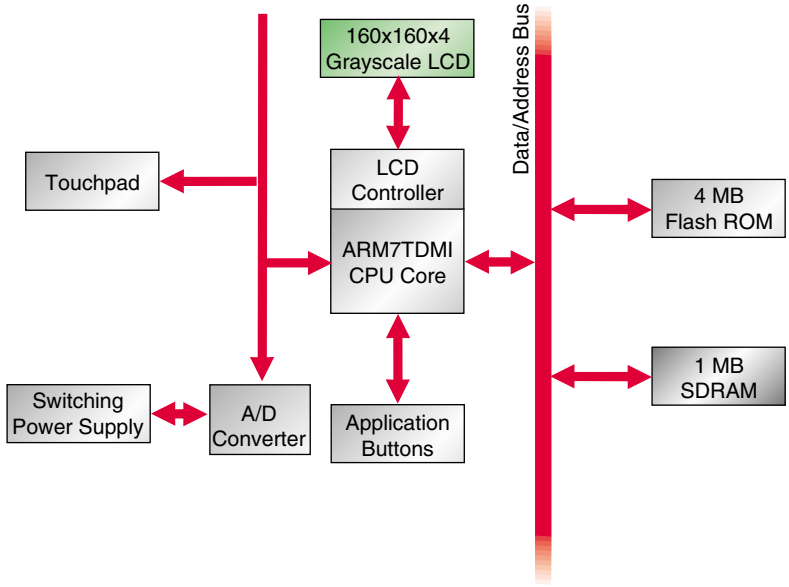
- Switching power supply and A/D converter

What is a Minimum System?

Figure 1.1 illustrated typical system configurations that are supported by Palm OS software release 5.0.

Figure 1.2 illustrates the minimum hardware configuration required to run Palm OS software release 5.0.

Figure 1.2 Minimal System Configuration



As shown, a minimal system requires a touchpad and display and some set of application buttons. A minimum of 4 MB of flash ROM is required (the Japanese ROM requires 8 MB). The operating system requires a minimum of 1 MB of SDRAM.

2

System Details

This chapter describes the characteristics of each of the major system components. The following major components are described:

- CPU
- ROM
- SDRAM
- LCD controller and display
- Touch screen and controller
- Battery
- Power supply
- IrDA and UART

CPU Specifications and Requirements

The Palm OS® 5 supports the ARM7 TDMI family of processors. The general capabilities of this processor are listed in Table 2.1.

Table 2.1 ARM7TDMI CPU Characteristics

Feature	Capabilities
LCD controller	16 gray levels
LCD resolution	160 x 16 or 320 x 320
Timer	Two general-purpose 16-bit timers with a programmable 8-bit prescaler
General Purpose I/O (GPIO) Ports	Maximum of 27 general purpose input/outputs

Table 2.1 ARM7TDMI CPU Characteristics *(continued)*

Feature	Capabilities
Serial Peripheral Interface (SPI)	Two synchronous serial interfaces for Micro-wire or SPI interfaces, one supporting both the master and slave, and the other supporting only master mode.
Universal Serial Bus (USB)	USB client interface compliant with USB Spec. Rev. 1.1
Digital Audio Interface (DAI)	Interfaces to low-power DACs, ADCs and CODECs. Supports MPEG and multiple audio decompression
PWM (for sound generation)	Two programmable pulse width modulation (PWM) interfaces
UART	Two full-duplex 16550A-compatible UARTs with a 16-byte transmit and receive FIFOs
Real Time Clock (RTC)	Low-power 32.768 KHz RTC
PCMCIA	PC card interface block on external bus
SDRAM controller	48 KB of on-chip SDRAM that can be shared between LCD controller and general application use
Package	208-pin LQFP, 256-ball PBGA, 204-ball TFBGA

CPU Requirements Overview

Although the operating system has some specific requirements concerning configuration of the CPU and connected hardware, you are given great latitude in the use of many of the ARM7TDMI CPU pins. For example, the assignment and usage of the general purpose I/O (GPIO) port pins is almost unrestricted, so long as the minimum requirements of the operating system are met.

Table 2.2 lists the signal names of the ARM7TDMI CPU. Note that signals that start with a small “n” (such as nRESET) are active low.

For more information on the usage of these signal names, consult Appendix A of the ARM7TDMI Rev 4 Technical Reference Manual in the Technical Documentation section of the ARM web site, www.arm.com.

Table 2.2 ARM7TDMI CPU Signal Names

Function Group	Signals	Usage
Power	VDD	Device power
Ground	VSS	GND reference
Clocks	MCLK nWAIT, ECLK	Main clocks and external clock output
Initialization	nRESET	System reset signal
Thumb state	TBIT	Thumb instruction set
Endianness	BIGEND	Big endian config
Address Bus/PFIO (A0-A23		
	A[31:0], nMREQ, SEQ, LOCK	32-bit address bus, not memory request, sequential address, locked operation
Data Bus		
	BL[3:0], DIN[31:0], DOUT[31:0], D[31:0], nRW, MAS[1:0]	Byte latch control, data input and output bus, data bus, not read or write, memory access size
MMU interface		
	nTRANS, nMODE[4:0], ABORT	Not memory translate, not processor mode, memory abort
Interrupt Control		

Table 2.2 ARM7TDMI CPU Signal Names *(continued)*

Function Group	Signals	Usage
	nIRQ, nFIQ, INSYNC	Not interrupt request, not fast interrupt request, synchronous interrupts
Bus Control		
	nENIN, nENOUT, nENOUTI	Not enable interrupt, Not enabled output, enable output internal
	ABE, ALE, APE, DBE, TBE,	Address bus enable, address latch enable, address timing pipeline enable, data bus enable, test bus enable
	BUSEN, HIGHZ, BUSDIS, ESCAPCLK	Data bus configuration, high impedance, bus disable, EXTEST capture clock
Debug		
	DBGRQ, DBGRQI, DBGACK, DBGEN, BREAKPT	Debug request, internal debug request, debug acknowledge, debug enable, break to enter debug state
	nEXEC, EXTERNO, EXTERN1	Not executed, external input 0, external input 1
	RANGEOUT0, RANGEOUT1, COMMRX, COMMTX	EmbeddedICE watchpoints 0 and 1, communications channel receive, transmit

Table 2.2 ARM7TDMI CPU Signal Names *(continued)*

Function Group	Signals	Usage
Coprocessor interface		
	nOPC, nCPI, CPA, CPB	Not op-code fetch, not coprocessor instruction, coprocessor absent, coprocessor busy
TAP information		
	TAPSM[3:0], IR[3:0], nTDOEN, TCK1, TCK2, SCREG[3:0]	TAP controller state machine, instruction register, not TDO enable, test clock phase 1 & 2, scan chain register
Boundary scan extension		
	DRIVEBS, ECAPCLKBS, ICAPCLKBS, nHIGHZ, PCLKBS, RSTCLKBS,	Boundary scan cell enable, EXTEST capture clock for boundary scan, INTEST capture clock, not HIGHZ, boundary scan update clock, boundary scan reset clock
	SDINBS, SDOUTBS, SHCLKBS, SHCLK2BS	Boundary scan serial input & output data, boundary scan shift clock, phases 1 & 2
JTAG		
	nTRST, TCK, TMS, TDI, TDO	Not test reset, test clock, test mode select, test data input & output

CPU AC Characteristics and Requirements

The ARM7TDMI CPU can operate over a fairly large frequency range (18 MHz to 1 GHz). The most commonly tested (and therefore recommended) frequency is 74 MHz.

The clock input to the CPU is fed into a *PLL* block internal to the processor chip and the PLL locks onto the input signal and generates a digital clock.

CPU Mechanical Specifications

The ARM7TDMI CPU is available in three packages: LQFB, PBGA, and TFBGA. The characteristics of these packages are listed in Table 2.3.

Table 2.3 CPU Package Specifications

Package	LQFB	PGBA	TFBGA
Area dimension (mm)			
Pins	208-pin	256-ball	204-ball
Height (mm)			
Pitch (mm)		1	

Non-Volatile (Flash) Memory Requirements

The recommended minimum flash memory size for a device based on Palm OS software release 5.0 is 4 MB. The operating system supports up to 16 MB of flash memory. This is the recommended size for devices supporting the Japanese language and for versions incorporating such code-intensive components as wireless communications.

Flash memory devices are recommended to enable updating of system software and applications. At this time, the only flash memory device supported by the Palm Universal Debugger (PUD) is the 3 Volt Intel StrataFlash Memory (J3), part number 28F128J3A. The ability to program other flash devices (and the instructions for adding support for other flash devices) will be available soon.

RAM Requirements

For information on RAM requirements see the "Memory Management" chapter of the *DAL Customization Guide*.

LCD Controller and Display

The ARM7TDMI CPU has a built-in LCD controller which supports up to 16 levels of gray. The internal controller does not require dedicated RAM, it uses the same memory as the CPU. Additionally, the internal LCD controller does not consume any CPU bandwidth because graphics data is transferred using DMA control. For these reasons, it is highly recommended that the internal LCD controller be used if the system is implementing a gray scale LCD display.

Palm OS software release 5.0 also supports up to 16-bit color on a 320x160 display. There are several considerations in choosing a color LCD controller:

- Operating voltage should be 3.3V to eliminate the need for an additional regulator.
- A controller with built-in display memory is preferred.
- The ability to work directly with an ARM processor is best, to eliminate the need to add any additional logic chips.

Battery and Power Considerations

Typical usage models for existing Palm connected organizers indicate that the devices spend 95% of the time in sleep mode and 5% in active mode. Maximum battery conservation is achieved by taking advantage of this usage pattern and reducing system power consumption during the idle or sleep time.

Table 2.4 lists the major system devices and their recommended state during different power modes.

Table 2.4 Device State vs. System Power Mode

Power Mode ->	Active	Sleep
CPU	On	Sleep mode
ROM (Flash)	On	On
RAM	On	Self-refresh by processor's DRAM controller.

Table 2.4 Device State vs. System Power Mode *(continued)*

RS-232	See RS-232 Interface Control	Off
IrDA	See IrDA Device Control	Off
Tablet	On	Off
LCD	On	Off
Power supply	On	On (3.3V only - others can be off)
ADC	On	Off
Internal clock	On	Off
RTC	On	On

Power Supply and Distribution

To utilize the maximum power, use a switching power supply. Depending on the number and type of cells, a boost and/or buck regulator will perform better than a linear regulator. Switching supplies typically can perform better than 80% and at up to 95% efficiency. Linear regulators can at best achieve 50% efficiency.

Voltages to the CPU and memory chips must be regulated. Voltages to the speaker and other devices can be tapped off the battery rail.

IrDA and RS-232 Interfaces

Both the IrDA and RS-232 interfaces can be connected directly to the ARM7TDMI CPU.

IrDA Device Control

A miniature transceiver module device is used for the IrDA interface. Typically, such a device consumes about 120 mA when it is powered on and is therefore an appreciable power drain. There are two ways to shut down the IrDA device to conserve power:

- completely remove power to the device
- put the device in a low-power shutdown mode via software control

While completely cutting power to the device results in zero current draw, in standby mode the device draws only 1 nA of current, a minimal difference. Therefore, the second method (use of the shutdown mode) is preferable because the transition to wake up the device from standby mode is much quicker than re-applying power to the device.

An additional consideration is that when enabling the IrDA device, the operating system expects it to be active within a stringent period of time. The time it takes to enable the device through applying power is not guaranteed and may result in loss of data.

For these reasons, the operating system requires that the IrDA be put in the low-power shutdown mode as opposed to being completely powered down.

A special consideration for the IrDA interface is when the device is set to automatically receive beamed information. This means that the interface is toggled in and out of shutdown mode regularly to check if there is an incoming IR signal.

RS-232 Interface Control

A voltage translator is required for the RS-232 interface since the ARM7TDMI CPU's operating voltage is 3.3V and the RS-232 specification requires ±15V.

A

Supported Flash Devices

The following devices (in alphabetical order by manufacturer name) are known to work with the Palm OS® 5. Although the software recognizes and supports all of them, the devices marked with an asterisk (*) have not been tested internally to certify that they flash properly and reliably. Note that in the manufacturer's data sheets, the manufacturer ID is often referred to as the Manufacturer Code, and the device ID is often referred to as the Device Code.

AMD: (manufacturer ID 0x01)

URL: http://www.amd.com/products/nvd/techdocs/techdocs.html

- AMD Am29LV800B
  - 8 Mbit
  - device ID 0x5B
- AMD Am29LV160BB
  - 16 Mbit
  - device ID 0x49
- AMD Am29DL323CB
  - Bottom boot block
  - 32 Mbit
  - device ID 0x53
- AMD Am29LV320DB
  - Bottom boot block
  - 32 Mbit
  - device ID 0xF9

- AMD Am29LV320DT
  - Top boot block
  - 32 Mbit
  - device ID 0xF6

Atmel: (manufacturer ID 0x1F)

URL: http://www.atmel.com/atmel/acrobat/doc0925.pdf

- Atmel AT49BV1614
  - Bottom boot block
  - 16 Mbit
  - device ID 0xC0

Fujitsu: (manufacturer ID 0x04)

URL: http://edevice.fujitsu.com/fj/DATASHEET/

- Fujitsu MBM29DL323BD
  - Bottom boot block
  - 32 Mbit
  - device ID 0x53
- Fujitsu MBM29DL323BE
  - Bottom boot block
  - 32 Mbit
  - device ID 0x53
- Fujitsu MBM29DL323TE *
  - Top boot block
  - 32 Mbit
  - device ID 0x50

Intel: (manufacturer ID 0x89)

URL: http://www.intel.com/design/flash/datashts/index.htm

- Intel i28F160B3B*
  - Bottom boot block
  - 16 Mbit
  - device ID 0xC3
- Intel i28F160B3T
  - Bottom boot block
  - 16 Mbit
  - device ID 0x91
- Intel i28F320C3B
  - Bottom boot block
  - 32 Mbit
  - device ID 0xC5
- Intel i28F320C3T
  - Bottom boot block
  - 32 Mbit
  - device ID 0xCC
- Intel E28F320J3A
  - Bottom boot block
  - 32 Mbit
  - device ID 0x16
- Intel E28F640J3A
  - Bottom boot block
  - 64 Mbit
  - device ID 0x17

- Intel E28F128J3A
  - Bottom boot block
  - 128Mbit
  - device ID 0x18

Macronix: (manufacturer ID 0xC2)

- Macronix MX29LV160B*
  - Bottom boot block
  - 16 Mbit
  - device ID 0x49
- Macronix MX29LV160T*
  - Top boot block
  - 16 Mbit
  - device ID 0xC7

Mitsubishi: (manufacturer ID 0x1C)

http://www.mitsubishichips.com/data/datasheets/memory/index.html#FlashMemory

- Mitsubishi M5M29FB800
  - Bottom boot block
  - 8 MBit
  - device ID 0x5E
- Mitsubishi M5M29GB320VP
  - Bottom boot block
  - 32 Mbit
  - device ID 0x21

- Mitsubishi M5M29GT320VP
  - Top boot block
  - 32 Mbit
  - device ID 0x20

Sharp: (manufacturer ID 0xB0)

URL: http://www.sharpsma.com/sma/products/memory_

- Sharp LH28F160BJE*
  - Bottom boot block
  - 16 Mbit
  - device ID 0xE9
- Sharp LH28F320BJHE
  - Bottom boot block
  - 32 Mbit
  - device ID 0xE3

Toshiba: (manufacturer ID 0x98)

URL: http://pdf.toshiba.com/taec

- Toshiba version of AMD Am29LV160BB
  - 16 Mbit
  - device ID 0x43
- Toshiba TC58FVB321FT-10
  - Bottom boot block
  - 32 Mbit
  - device ID 0x9C

ARM HRS Glossary

ADC Analog to Digital Converter.

AHB Advanced High-Performance Bus.

AMBA Advanced Microcontroller Bus Architecture, created by ARM, Inc.

ARM Advanced RISC Machine; Acorn RISC Machine.

ARM720T A 32-bit microprocessor based on the ARM7TDMI RISC CPU core.

ASB Advanced System Bus.

BGA Ball Grid Array.

CCFT Cold Cathode Florescent Tube. Used as a light source for a backlit screen. Weighs more and uses more power than other back lights.

CODEC Coder Decoder. Hardware and software that converts sound, speech, or video from analog to digital, and vice versa.

Cotulla Microprocessor "system on a chip" based on the Intel XScale™ architecture.

DAI Digital Audio Interface.

DRAM Dynamic Random Access Memory. Requires refresh circuitry.

DSP Digital Signal Processing.

FPGA Field-Programmable Gate Array logic chip.

FSTN	Film Compensated STN. A passive matrix LCD.
Harvard architecture	Simultaneous access to instruction and data in memory. Compare to von Neumann architecture.
HEK	Hardware Evaluation Kit for Palm OS Software.
IrDA	Infrared standard used in Palm devices for beaming information.
JTAG	Joint Test Action Group. Software environment and interface for debugging units such as MultiICE and JENNI to gain direct access to the ARM processor.
JTEK	Java Technology Enabling Kit. Provides Jazelle support code for ARM.
LQFP	Low-profile Quad Flat Pack.
MAPBGA	Mold Array Process Ball Grid Array. See BGA.
MMU	Memory Management Unit.
Multi-ICE	Multi-processor Embedded In-Circuit Emulator for ARM. Enables non-intrusive debug of embedded processors at full speed.
PBGA	Plastic Ball Grid Array.
PLL	Phase Locked Loop. A technique for maintaining synchronization in an electronic circuit.
PWM	Pulse Width Modulator.
RTC	Real Time Clock.
SPI	Serial Peripheral Interface.
STN	Super Twisted Nematic. A type of LCD.

TAP	Test Access Point for Multi-ICE.
TCM	Tightly-Coupled Memory. An alternative approach to caches, designed to enable high-performance operation when attached to slow external memory.
TDMI	ARM Thumb architecture extension; Debug extension core, 32x8 Multiplexer, and EmbeddedICE logic extensions.
TFBGA	Thin Fine-pitch Ball Grid Array.
TFT	Thin Film Transfer, such as the active matrix screen on a handheld display.
TLB	Translation Look-aside Buffer. A table used in a virtual memory system that lists the physical address page numbers associated with each virtual address page. Used in conjunction with a cache whose tags are based on virtual addresses.
TQFP	Thin Quad Flat Pack. A square, surface-mount chip package with leads on all four sides.
TSOP	Thin Small Outline Package. Surface mount chips with gull-wing pins on the two short sides.
UART	Universal Asynchronous Receiver/Transmitter. Used to communicate with external serial devices. Serial port.
VF BGA	Very thin profile Fine pitch BGA.
von Neumann architecture	A load/store architecture using one 32-bit data bus for both instructions and data in a sequential access. Only load, store, and swap instructions can access data in memory. Compare with Harvard architecture.
X Scale	Intel ARM chip with V5TE compatibility.